

Знакомство с Matlab

Храмов Д. А.

18.03.2020

В этой лекции

- ▶ Что такое “научные вычисления”.
- ▶ Matlab: первое знакомство, преимущества и недостатки.
- ▶ Где искать информацию по курсу.

Научные вычисления

Научные вычисления (scientific computing, computational science)
— решение научных задач с помощью компьютеров.

- ▶ Задачи чаще всего относятся к области естественных наук (физики, химии, биологии и др.) или к математике.
- ▶ Научные вычисления называют “третьим столпом науки”, наряду с теоретическим анализом и экспериментами.
- ▶ Грань между научными вычислениями и науками, являющимися источниками задач, проходит между интересом к методам решения задач и решением самих задач.

Компьютер позволяет выполнять расчеты:

- ▶ быстрее;
- ▶ дешевле;
- ▶ иным способом, чем это было в “докомпьютерную эпоху”.

Как это делается

Научные вычисления предполагают:

1. создание математической модели — приближенного описания явления на языке математики,
2. реализацию модели на языке программирования,
3. выполнение расчетов,
4. проверку правильности результатов.

Для этого нужно знать:

- ▶ **Математику**
- ▶ Программирование
- ▶ Вычислительные методы
- ▶ Математическое моделирование

Matlab



MATLAB (MATrix LABoratory) придуман Кливом Молером (Cleve Moler) в конце 1970-х гг. чтобы работать с библиотекой LINPACK, не изучая Fortran. Фото 1977 г.

Сложение матриц: C

```
#include <stdio.h>
#define N 3

int main()
{
    int A[N][N] = { {1, 1, 1},
                    {2, 2, 2},
                    {3, 3, 3}};
    int B[N][N] = { {1, 1, 1},
                    {2, 2, 2},
                    {3, 3, 3}};

    int C[N][N];
    int i, j;

    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
            C[i][j] = A[i][j] + B[i][j];

    // Print result here...

    return 0;
}
```

Сложение матриц: Matlab

```
A = [1, 1, 1;  
     2, 2, 2;  
     3, 3, 3];  
B = [1, 1, 1;  
     2, 2, 2;  
     3, 3, 3];
```

A+B

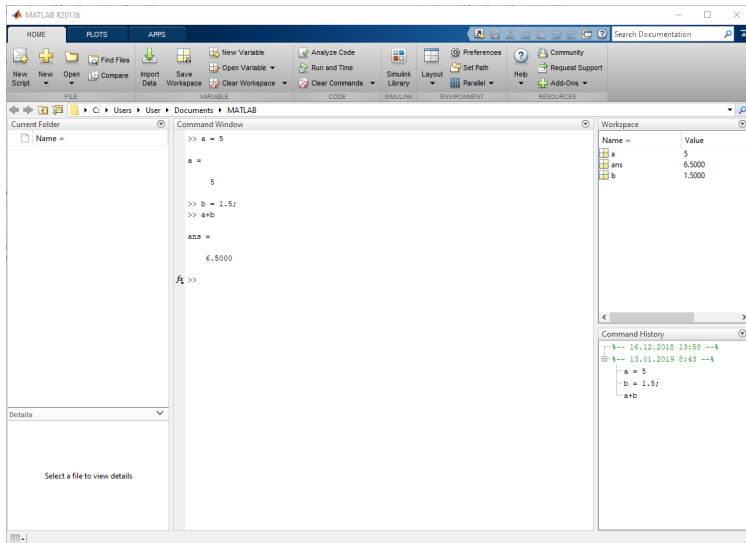
- ▶ нет директив препроцессора
- ▶ нет описания типов данных
- ▶ нет циклов
- ▶ есть вывод

А как со скоростью?

Пример неэффективного (но рабочего) кода Matlab

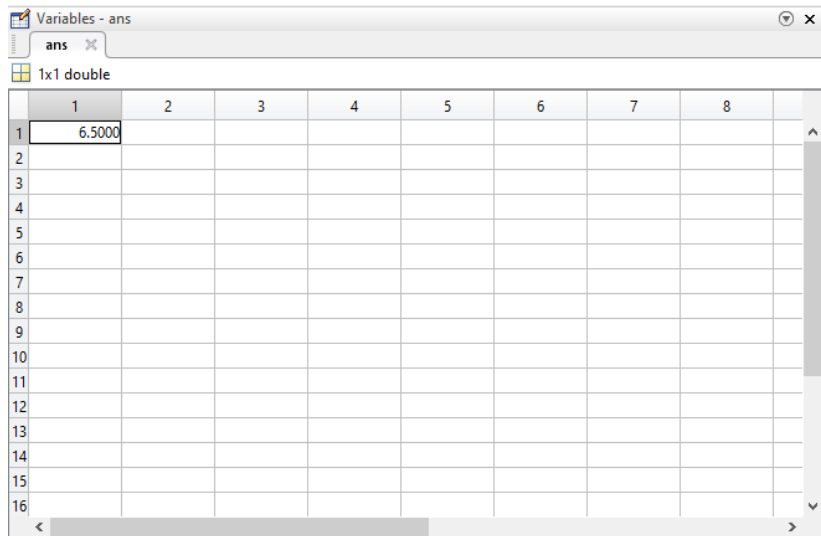
```
A = [[1, 1, 1];  
      [2, 2, 2];  
      [3, 3, 3]];  
B = [[1, 1, 1];  
      [2, 2, 2];  
      [3, 3, 3]];  
C = zeros(size(A));  
  
for i = 1:3  
    for j = 1:3  
        C(i,j) = A(i,j) + B(i,j);  
    end  
end  
  
C
```


Интерфейс Matlab



- ▶ **Командное окно (Command Window)** — набираем команды и нажатием *Enter* отправляем их на выполнение.
 - ▶ *Shift-Enter* — перевод строки
 - ▶ `clc` — очистка командного окна
 - ▶ `↑`, `↓` — перемещение по истории команд
- ▶ **Рабочая среда (Workspace)** — просмотр содержимого памяти.
 - ▶ `clear` — очистка содержимого памяти
- ▶ **История команд (Command History).**

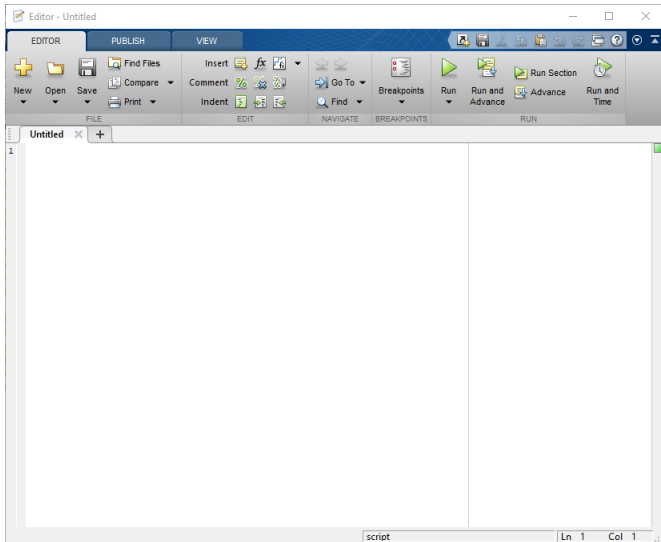
Рабочая среда (Workspace)



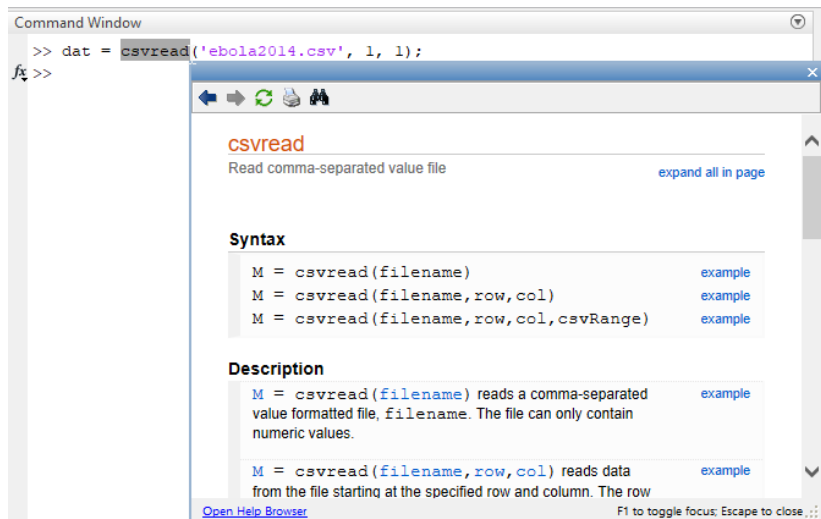
The image shows a MATLAB 'Variables' window titled 'Variables - ans'. It contains a single variable named 'ans' which is a '1x1 double' with the value 6.5000. The variable is displayed in a table with 16 rows and 9 columns. The first row is highlighted, showing the value 6.5000 in the first column. The rest of the table is empty.

	1	2	3	4	5	6	7	8
1	6.5000							
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

Ctrl+N - вызываем редактор



F1 - Справка



The screenshot shows the MATLAB Command Window with the command `>> dat = csvread('ebola2014.csv', 1, 1);` entered. A help window for the `csvread` function is open, displaying the following information:

csvread
Read comma-separated value file expand all in page

Syntax

<code>M = csvread(filename)</code>	example
<code>M = csvread(filename, row, col)</code>	example
<code>M = csvread(filename, row, col, csvRange)</code>	example

Description

<code>M = csvread(filename)</code> reads a comma-separated value formatted file, <code>filename</code> . The file can only contain numeric values.	example
<code>M = csvread(filename, row, col)</code> reads data from the file starting at the specified row and column. The row	example

[Open Help Browser](#) F1 to toggle focus; Escape to close

ПРИМЕР. Данные по смертности от лихорадки Эбола

Файл ebola2014.csv:

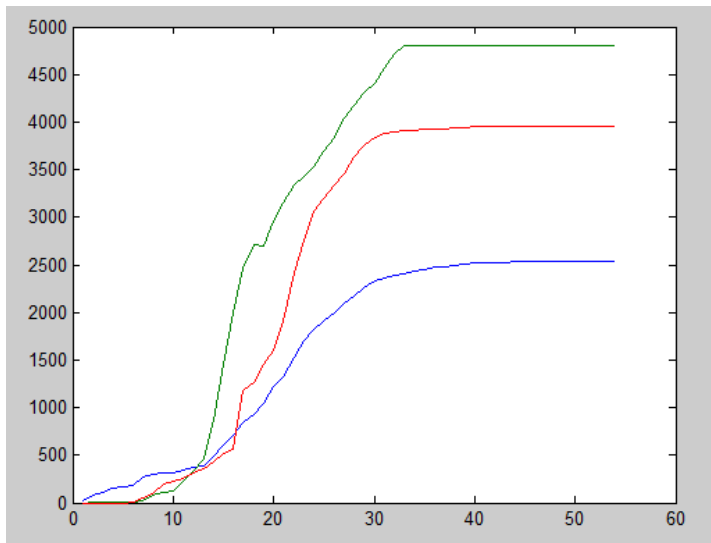
```
Date,Guinea,Liberia,Sierra.Leone
2014-03-22,29,0,0
2014-03-31,80,2,0
2014-04-14,108,2,0
2014-05-01,149,11,0
...
```

Скрипт plot_graph.m:

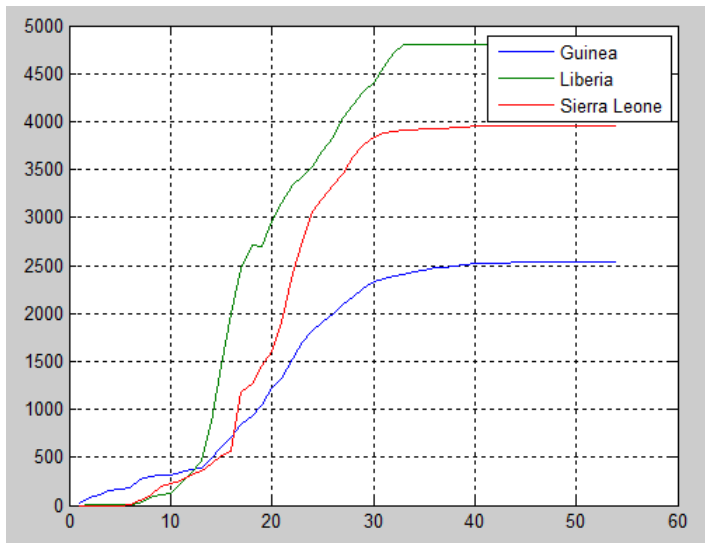
```
dat = csvread('ebola2014.csv', 1, 1);
plot(dat)
```

Данные: https://en.wikipedia.org/wiki/West_African_Ebola_virus_epidemic_timeline_of_reported_cases_and_deaths

Запустим скрипт



Украсим результат



Переменные

`a = 3.5 % Переменная = Значение[;]`

`b3 = 1`

`A = 6`

`','` – ставят после промежуточных вычислений для подавления вывода в командное окно или для разделения команд в одной строке.

Язык Matlab – РЕГИСТРОзависимый.

`ans` (от *answer*) – имя “анонимной” переменной.

`%` – начало комментария.

Синтаксис переменных:

- ▶ латиница, цифры, знак подчеркивания,
- ▶ начинается с буквы,
- ▶ кроме зарезервированных слов (`for`, `end`, `if`, `else`, `case`,...).

Внимание! Правила относительно имен переменных распространяются на функции, а также на файлы и папки.

Типы данных

Все числовые переменные по умолчанию имеют тип `double` – 8 байт. Поэтому нет необходимости указывать тип переменных. Однако в языке есть функции для явного приведения типов.

Другие типы данных:

- ▶ Строковые: `'Hello world'`, `'Aren't you crazy?'` — выражение в одинарных кавычках является строковой переменной.
- ▶ Комплексные числа: `z = 3+6i`.
- ▶ `int*`, `uint*` — целые числа полезны при работе с изображениями.
- ▶ ...

Контейнеры:

- ▶ массивы, структуры (`structure`), ячейки (`cell`), таблицы (`table`).

Узнать тип данных A: `class(A)`

Матрицы

- ▶ Все числовые переменные в Matlab — это матрицы (массивы).
- ▶ Matlab = MATrix LABoratory.
- ▶ Скаляр — матрица размерности 1x1.

```
a = [1 2 3 4 5 6 7 8 9]    % матрица 1x9
b = [1,2,3,4,5,6,7,8,9]'    %           9x1
A = [1 2 3; 4 5 6; 7 8 9]    %           3x3
```

- ▶ Матрицы записываются по строкам
- ▶ Строки разделяются “;”
- ▶ Столбцы разделяются пробелами или “,”

```
A = [1 2 3;
      4 5 6;
      7 8 9]
```

Обращение к элементам

- ▶ Нумерация элементов начинается с 1.

```
a = [1 2 3 4 5 6 7 8 9]
a(1)      % первый элемент
a(end)    % последний элемент
a(3:5)    % элементы с 3-го по 5-й
a([5,7])  % элементы 5-й и 7-й
```

- ▶ двоеточие : задает диапазон изменений — с:по

```
x =  начало:шаг:конец;

a = 1:9      % 1 2 3 4 5 6 7 8 9
b = 1:3:9    % 1 4 7
c = 9:-2:0   % 9 7 5 3 1
```

Поэлементные операции

Matlab трактует $*$, $/$, $^$ как операции, выполняемые по правилам линейной алгебры.

```
x = [1 2 3];  
y = [3 4 5];  
x * y
```

Дает:

```
Error using *  
Inner matrix dimensions must agree.
```

Для выполнения операций над отдельными элементами матриц нужно поставить перед знаком нужной операции точку: $.*$, $./$, $.^$

```
x .* y
```

```
ans =
```

```
3     8    15
```

Приоритет операций

Операция	Описание
()	изменяют приоритет операций
'	транспонирование
^	возведение в степень
.^	поэлементное возведение в степень
\	матричное деление
*, /	умножение, деление
.*, ./	поэлементное умножение, поэлементное деление
+, -	сложение, вычитание

Элементарные функции

`sin(x)`, `cos(x)`, `tan(x)`, `cot(x)`,
`sqrt(x)`,
`exp(x)`, `log(x)`, `log10(x)`,
`sinh(x)`, `cosh(x)`, `tanh(x)`

- ▶ Аргументы тригонометрических функций задаются в радианах.
- ▶ Найти x радиан по заданным d градусам можно с помощью пропорции: $x/d = \pi/180$, т.е. $x = d \cdot \pi/180$.
- ▶ Аргументы тригонометрических функций `cosd()`, `sind()` задаются в градусах.

ПРИМЕР. Поиск корней уравнения

Найдем корень уравнения

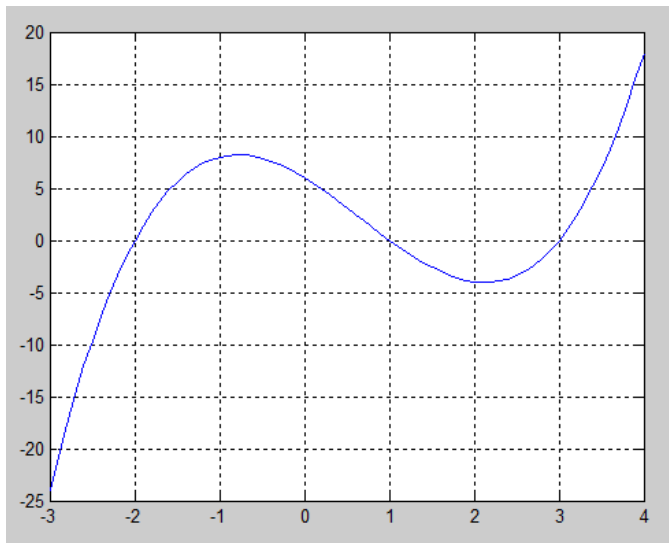
$$x^3 - 2x^2 - 5x + 6 = 0,$$

расположенный на промежутке $[0; 2]$ с точностью до 0.01.

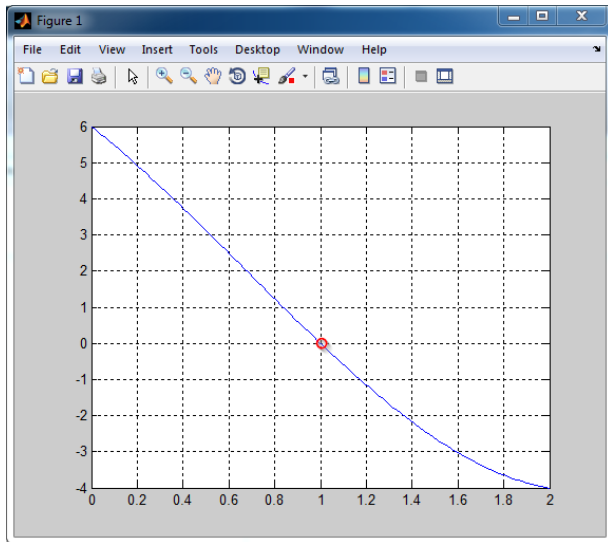
Строим график

```
x = -3:.1:4;  
y = x.^3 - 2*x.^2 - 5*x + 6;  
plot(x,y), grid on
```

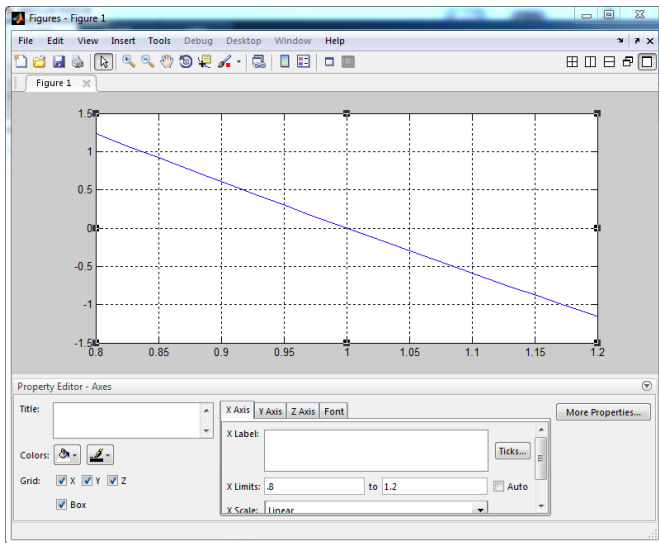
- ▶ Попробуйте построить график с шагом по x , равным 1.
- ▶ Посмотрите, как выглядит матрица x в *Workspace*.



Ограничиваем область поиска корня: меню Edit/Axes Properties...



Сужаем окрестность корня (XLimits) пока не достигнем нужной точности



При какой длине интервала по x мы достигнем заданной точности?

format: представление чисел в командном окне

```
Command Window
>> 4/3

ans =

    1.3333

>> format long
>> 4/3

ans =

    1.3333333333333333
```

Специальные значения

Бесконечности:

$$\text{Inf} = 1/0$$

$$-\text{Inf} = -1/0$$

Not-a-Number:

$$\text{NaN} = 0/0$$

Математические константы:

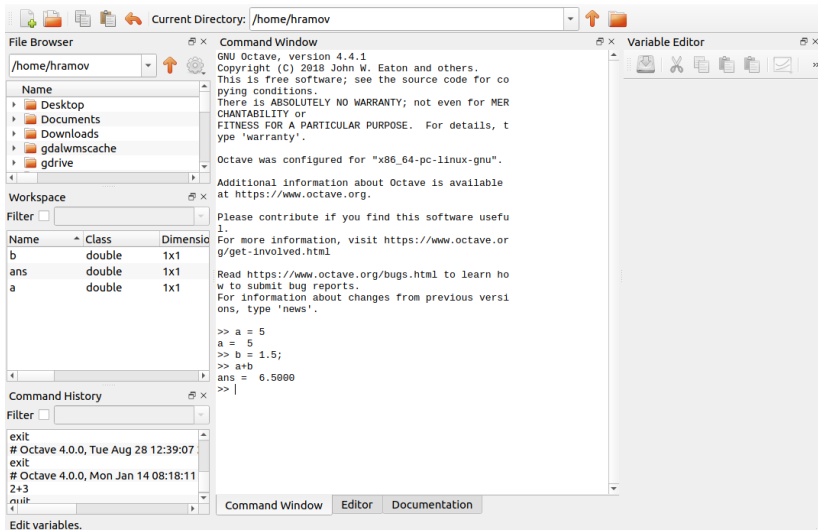
π

$$e = \exp(1)$$

Бесплатная альтернатива Matlab

- ▶ GNU Octave — ближайший аналог
- ▶ Scilab
- ▶ Python + NumPy + SciPy
- ▶ Julia

Интерфейс GNU Octave



Online

- ▶ Matlab Online: <https://matlab.mathworks.com/> (бесплатная регистрация)
- ▶ Octave Online: <https://octave-online.net> (без регистрации)

Настольные

- ▶ > Matlab R2013b — работает на Windows 7 и Windows 10 (вот как она выглядит:
<https://rutracker.org/forum/viewtopic.php?t=4554638>)

Информационные ресурсы

1. Ануфриев И.Е. и др. MATLAB 7.
2. Хант Б. и др. MATLAB R2007 с нуля!
3. MATLAB: язык технических вычислений.
4. matlab.exponenta.ru
5. dkhramov.dp.ua: m-файлы примеров

Вспоминаем математику

- ▶ Бёрд Дж. Инженерная математика: Карманный справочник. М.: Издательский дом «Додэка-XXI», 2008. 544 с.

Где искать книги

- ▶ Папка /lit
- ▶ Library Genesis: gen.lib.rus.ec

Контактная информация

Преподаватель: Храмов Дмитрий Александрович, доцент, к.т.н.

e-mail: `dkhramov@mail.ru`

веб-сайт: `dkhramov.dp.ua`

skype: `d_khramov`

GitHub: `https://github.com/dkhramov`